

Reducing Reauthentication Delay in Wireless Networks

Tuomas Aura and Michael Roe
Microsoft Research, Cambridge, UK
{tuomaura,mroe}@microsoft.com

Abstract

When a wireless mobile user is moving across a mobile network or between co-operating networks, the network operators often want to verify the user's access rights before granting service. The security protocol causes a delay in the network access, which may be much longer than the typical delays caused by mobility management. An alternative would be to provide so called optimistic service before the user has been authenticated or paid for the access. Thus, there is a trade-off between the security of the access control and the quality of service observed by the user. Our aim is to reduce the authentication delay and to enable optimistic access without opening a window for fraudulent access. We present a protocol for the reauthentication of a mobile node when it repeatedly connects to different access points or co-operating wireless networks. The protocol is based on credentials which the mobile receives from access points as a proof of past honest behavior and which it presents when associating with a new access point. It can be implemented with keyed one-way functions that result in low computation and communication overhead both for the mobile and for the network.

1. Introduction

Interactive data connections, live video, and multimedia are often seen as the core applications that drive the construction of future mobile networks. The promise to the user is to be able to access the same services everywhere and to move seamlessly between access points and between wireless networks. In this paper, we mainly consider networks based on the 802.11 technology [8] but the same ideas may be applicable to cellular data connections and to other network technologies. A major technical challenge in providing such services is the variable quality of service (QoS) of the mobile network connection. Network latency can make interactive services unusable and high variations in the delay (jitter)

create problems for real-time services such as video streaming.

Access control mechanisms can be one of the major sources of latency and jitter. When a mobile node moves between network areas, between networks belonging to different operators, or between different media types, some data packets usually get lost or take longer to arrive at the receiver. Mobility protocols that pay attention to QoS try to perform location updates as locally as possible and avoid message exchanges with remote nodes that might not be available at the time. Security protocols, on the other hand, typically require contacting a remote server [7][12][13]. In the case of authentication and authorization, messages are exchanged with a trusted on-line authority or a certificate revocation database. In on-line payment, a connection to a bank, credit agency or broker is needed. Many security protocols also require computation of complex cryptographic functions, which can be slow on low-power mobile devices. Figure 1(a) illustrates the authentication delay between the mobile node's (MN) arrival to the access network and the time when the access is authorized.

What is then the reason for invoking the security protocols in a wireless network? Firstly, the wireless LAN security protocols [6] are designed to authenticate the mobile user at each access point. Any building larger than a single-family house will usually have multiple access points and the mobile will need to move between them and to authenticate to each one. Secondly, if network operators co-operate to enable roaming across multiple networks in a larger area, such as a campus, office building or town, each network must authenticate the mobile to check that it has the right to use the service. Finally, commercial access-network operators want to ensure that the mobile users are authorized customers and that a payment for the service has or will be received. Therefore, they need to verify the user identity or authorization before granting access. Telecom operators are traditionally very keen on preventing unauthorized access. For them, unpaid seconds or bits equal lost revenue. Thus, when the mobile node

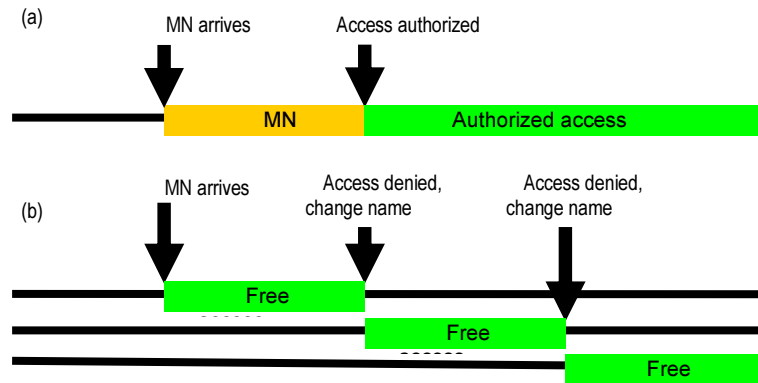


Figure 1 (a) authentication delay (b) misuse of optimistic access

moves across almost any type of wireless networks, connecting to one access point after another, it must be authenticated at every access point.

One solution is so called *optimistic access*, which means allowing access first and checking the authorization later. Optimistic access fits the thinking of wired network operators who rely on the user's location for authorization and, thanks to flat-rate charging, rarely need to consider accounting and fraud prevention. Mobile access, on the other hand, is typically charged by the connection time or transferred bits because this fits existing business models and it is a way of regulating the use of the limited wireless bandwidth. Optimistic access does not work well together with such charging schemes. The small window of unauthenticated access can be misused to transfer short messages to the Internet. As shown in Figure 1(b), a user with a modified protocol stack that changes the mobile name and hardware address after each failed authentication can repeatedly take advantage of the window of free access to obtain virtually continuous free connectivity. Even if only a minority of the users has the technical sophistication to take advantage of the vulnerability, most commercial network operators are unlikely to accept such a vulnerability. While it would be possible to allow optimistic access for the time being and to disable it when problems arise, that could result in a sudden drop in service quality after the users already have grown used to the higher QoS. Thus, current network technologies do not support optimistic access.

We would like to find a protocol that allows optimistic access but prevents or reduces the risk of misuse. Designing such a protocol is the goal of this paper. We recognize the fact that a strong authentication and verification of access rights has its cost. A light-weight protocol that avoids that cost

must relax some of the security requirements. We therefore look into an idea that was put forward by Meadows [10] in the context of denial-of-service prevention: start with a relatively weak but inexpensive authentication and perform a stronger and more costly one only after the weak one has succeeded. After the weak protocol, the server has some level of assurance that the client is honest and can allocate some resources to it. In our case, we allow optimistic access after the relatively weak protocol and delay strong authentication until later.

The main idea of our protocol is to reward frequent and well-behaving customers. A mobile that has previously demonstrated its willingness to pay for the service or to follow other rules is given optimistic access while an unknown mobile node will have to wait for the strong authentication. This way, the access will only be delayed when the mobile visits a network or a group of co-operating networks for the first time. The technique we use is a *cryptographic credential*, which is given to the mobile node after each strong authentication or payment and which it presents when it moves to a new network and wants instant access.

Throughout the paper, we use a mixed terminology borrowed from several mobile networking protocols. The mobile user device is called a *mobile node* and the support nodes via which it connects to the Internet are called *access points*. While we are obviously thinking of 802.11 wireless LANs, the protocol does not depend on a specific mobile networking technology. The protocol should be general enough so that it can be applied in various situations where the same mobile node repeatedly accesses the same network or multiple co-operating networks.

The closest existing technology to our protocol is the 802.1x pre-authentication in wireless LANs,

which is expected to become common with the 802.11i standard. It allows the mobile to authenticate to the (potential) next access point before associating with it and before disassociating from the previous access point. The pre-authentication only works when the mobile can guess which access point it will next use and when the old and new access points are on the same wired link. Our protocol is independent of the wired network topology and it can be used for reauthentication across LANs and network operators. Another existing solution is to move the authentication (e.g., RADIUS authenticator) from the access point to a *wireless switch* that controls several access points. This allows the mobile to move freely between the small group of access points without reauthenticating.

In Section 2, we describe the protocol requirements. Sections 3 and 4 outline the reauthentication protocol and the access credentials, respectively. Section 5 contains the complete protocol and Section 6 discusses the trust parameters, i.e., authorization information conveyed by the credentials. The efficiency and security of the protocol are analyzed in Sections 7 and 8. Section 9 concludes the paper.

2. Reauthentication requirements

In addition to the main goal of reducing the authentication delay for the reauthentication, there are several additional requirements for the protocol that are imposed by the mobile network environment. For example, mobile nodes often have severely limited computational power, memory and communications bandwidth. Security protocols for such devices must take these limitations into account. The requirements for our protocol design are listed below.

Fast service for recent users: The main goal of the protocol is to reduce authentication delay for mobile users who have recently accessed the same network. The access point must be able to verify that it or another access point has previously provided service to the same mobile and that the mobile has behaved in an honest manner. The honest behavior usually means successful authentication or payment. In our protocol, the access points will give the mobile credentials to use as a proof of its past honest behavior, and the mobile user must not be able to produce false credentials.

Security proportional to the threat: We are not suggesting to replace existing strong authentication protocols with ours. Instead, the aim is to design a protocol that is strong enough to authorize optimistic access for a short time period (usually seconds) until the strong authentication has taken place. By breaking our protocol, the malicious mobile will only be able to

misuse the optimistic access period to gain free Internet access. This is a relatively low threat and we can trade security for efficiency. (The main trade-off we will make is to take the risk of one mobile distributing its credentials to others.)

Low communication overhead: The number of bits and, in particular, the number of messages transferred between the protocol parties should be minimized. First, the communications bandwidth on the wireless link is limited and must not be consumed by excessive signaling. Second, connections between the access point and any on-line authorities should be avoided because they are time-consuming and may be unreliable. Our goal is to postpone all online contacts to the authorities until later and to allow network access after a brief local message exchange between the mobile and the access point. Use of secret-key cryptography helps in conserving bits on the radio interface.

Low computational overhead: Mobile devices have limited processing power due to their small physical size and battery capacity (e.g., handheld computer or a smart card on a mobile device). The computation required by the access control protocol must not cause significant delays or create new limitations on the frequency of location updates. Therefore, public-key cryptographic operations should be avoided. Symmetric (secret-key) operations like one-way hash functions and encryption can be used but the number of operations should be minimized.

Independence of user identifiers: The protocol should not rely on any specific type of identifier for the mobile user. Although the protocol in some applications may involve verification of the user identity, it should not be a required feature. On the contrary, the mobile should be able to prove its right to access the network without revealing its identity. For example, the reauthentication protocol should not prevent the implementation of anonymous prepaid user accounts. The original authentication by the first access point may or may not involve identity authentication and, similarly, the credential given to the mobile may or may not contain information about its identity. That way, our protocol can be combined with various anonymity protection techniques such as anonymous payment systems [5], as well as with identity-based PKIs [4].

Denial-of-service (DoS) resistance: The protocol should not create new opportunities for DoS attacks against the mobile network or against individual mobile users. While DoS in a wireless network is always more or less possible, the protocol should be designed so that an attacker cannot consume large amounts of processor, memory or communications capacity without its own constant

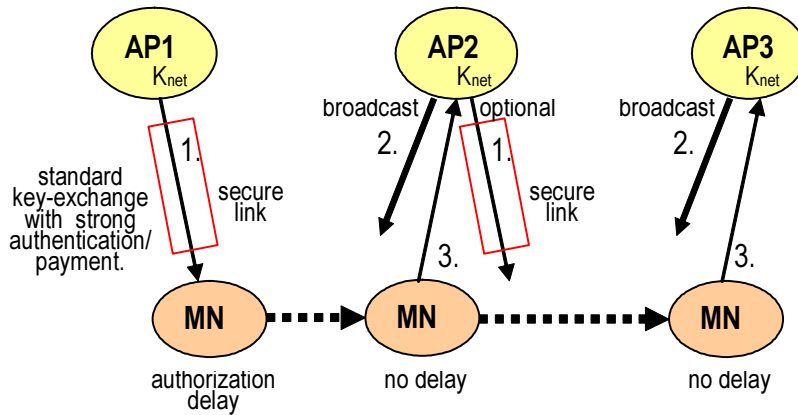


Figure 2: Mobile node moving between access points

and equally resource-consuming involvement. (The relation of the cost of attack to the created damage has been used as a definition of DoS. See Meadows [10] and Aura et al. [1].) It is also an advantage if the attacker cannot selectively target individual users but has to either target random nodes or block all traffic in the network. Anonymity protection usually has this as a side effect.

3. Protocol outline

In our protocol, all communication takes place between the mobile node (MN) and the access points (AP) over the wireless link. The access points do not communicate directly with each other. Instead, all information between them is communicated via the mobile node. Nevertheless, the wireless communications overhead of our protocol is relatively small. The 802.11 wireless LAN link-layer and most network-layer mobility protocols require the mobile node to exchange messages with the new access point when updating its location. The data of our protocol can be piggybacked on these messages at little additional cost. This means that the number of messages sent over the wireless interface need not increase.

It should be noted that some mobility protocols allow the mobile to negotiate the handover purely with the old access point. Mobile telephone systems, like GSM [7], can operate like this because the connection-oriented communication, for which they have been designed, requires uninterrupted service and coverage areas of the old and new access point always overlap. In our scenario, the mobile may move outside the network coverage and attach to a new access point after the connection with the previous one has been lost. Hence, the location update protocol has to involve a direct message exchange with the new access point.

Figure 2 outlines the messages of our protocol. The mobile node receives a credential from an access point AP1, where it has been successfully authenticated or paid for the access. AP1 gives the mobile a credential as a proof of its verified honest behavior. When the mobile arrives at a new access point AP2, it presents this credential to AP2. It also uses a cryptographic mechanism to prove that it is indeed the same mobile to which the credential was issued. This prevents the mobile from using pirated credentials that were issued to other mobiles. We propose using a challenge-response protocol for the proof.

While the challenge message could be avoided by assuming synchronized clocks at the mobile and the access point, the increased security of the challenge-response scheme is in most cases worth the small increase in cost. With careful design, the access point can broadcast challenges periodically, and the same challenges can be shared by all mobiles. In most mobility protocols, the access points advertise themselves with regular broadcasts of beacon messages, on which the broadcast messages of our protocol can be piggybacked.

In addition to the challenge-response exchange, the mobile node and AP2 need to agree on a secret key. In a way, we want to transfer a security association that exists between MN and AP1 to be between MN and AP2. It is possible to either produce a new key for MN and AP2 or to send the old key encrypted. We have opted for the former.

If the mobile continues its movement and connects to yet another access point, it may either use the credential from AP1 or it may have received a new credential from AP2. This depends on the expiration times of the credentials and on the policies that the access points have for issuing and accepting them. The protocol itself puts few restrictions on the choice of these policies.

The contents of the protocol messages are the following:

1. AP1→MN: K, Credential (sent over a secure link)
2. AP2→MN: i, N (broadcast challenge)
3. MN→AP2: i, $f_K(N, \text{macaddr}_{MN}, \text{macaddr}_{AP2}), \text{Credential}$

In the first message, the old access point gives the mobile node a secret value K (e.g., a fresh random bit string) and the credential. The purpose of the credential is to tell other access points that anyone who knows the secret K should be trusted for optimistic network access. Some time after receiving Message 1, the mobile moves to the coverage area of another access point. The second and third messages are a challenge-response exchange for verifying that the mobile node knows K. In Message 2, the new access point broadcasts a random number N. The mobile computes and sends a keyed one-way function f of the secret K and the challenge N. It also passes the credential to the new access point. The new access point verifies the credential, recovers the key K from the credential (the details will be explained below), and verifies the response. In practice, it may accept responses to the two or three latest broadcast challenges. i is a challenge sequence number that makes it easier to match the challenges and responses. It has no effect on the security of the protocol. The MAC address of the mobile's network interface is included in the input of the one-way function to mitigate some threats that will be discussed in Section 8.

After exchanging messages 2 and 3, the mobile and the new access point compute a session key that will be used to protect (encrypt and/or authenticate) data between them until the strong authentication has taken place. The protection of data is important not for the sake of the data itself but to bind the network access during the optimistic access period to the reauthentication and to the credential. It prevents another mobile from hijacking the session during the optimistic access. The session key is computed as follows:

$$K_{\text{ses}} = w_K(N, \text{macaddr}_{MN}, \text{macaddr}_{AP2})$$

w is a keyed one-way function that is different from f. Thus, the session key is a one-way function of the secret key K, the new access point's nonce N, and the mobile's MAC address.

4. Credential structure

As a first attempt, the credential could be implemented as a signed and encrypted message that contains the secret key K and binds it to the level of trust that the mobile should be shown.

$$\text{Credential} = E(S(K, TS, \text{trust params}))$$

The credential also contains a time stamp TS, which requires a level of clock synchronization between access points. Since the credential is created and read only by the access points and the mobile simply passes it unaltered to the next access point, the mobile does not need to know about the format of the credential. The parameters *trust params* can contain any information about the mobile that the old access point wants to pass to the new one. The credential may, for example, contain the time and type of the previous strong authentication and verification of access rights or payment. These parameters are further discussed in Section 6.

The encryption is needed to hide the secret key K. The signature and encryption must be implemented in such a way that any one of the co-operating access points and networks can verify them. This means that the decryption key needs to be shared by the access points. For this reason, we use shared-key cryptography and, since the shared key is needed anyway, a keyed one-way function instead of a public-key signature. The next (still not final) version of the credential is

$$\text{Credential} = j, E_{K_{\text{net}}}(K, TS, \text{trust params}, g_{K_{\text{net}}}(K, TS, \text{trust params})).$$

E denotes symmetric encryption and g is a keyed one-way function. K_{net} is the *network key*, a secret key shared by all access points. The index j identifies the key K_{net} . This index is useful during network-key updates and if there are multiple overlapping coalitions of access points or networks. (To avoid accidental collisions and the need for coordinating the assignment of j values, the index should be a randomly chosen 64-bit number.) The symmetric cryptography makes the verification of the credential fast and the messages short in comparison to public-key signatures and encryption.

We can, however, do even better by implementing the credential with only keyed one-way functions. In order to avoid encryption, we compute that secret K as a keyed one-way function of the network key K_{net} and a random (but not secret) nonce N_{AP1} . Thus, we define

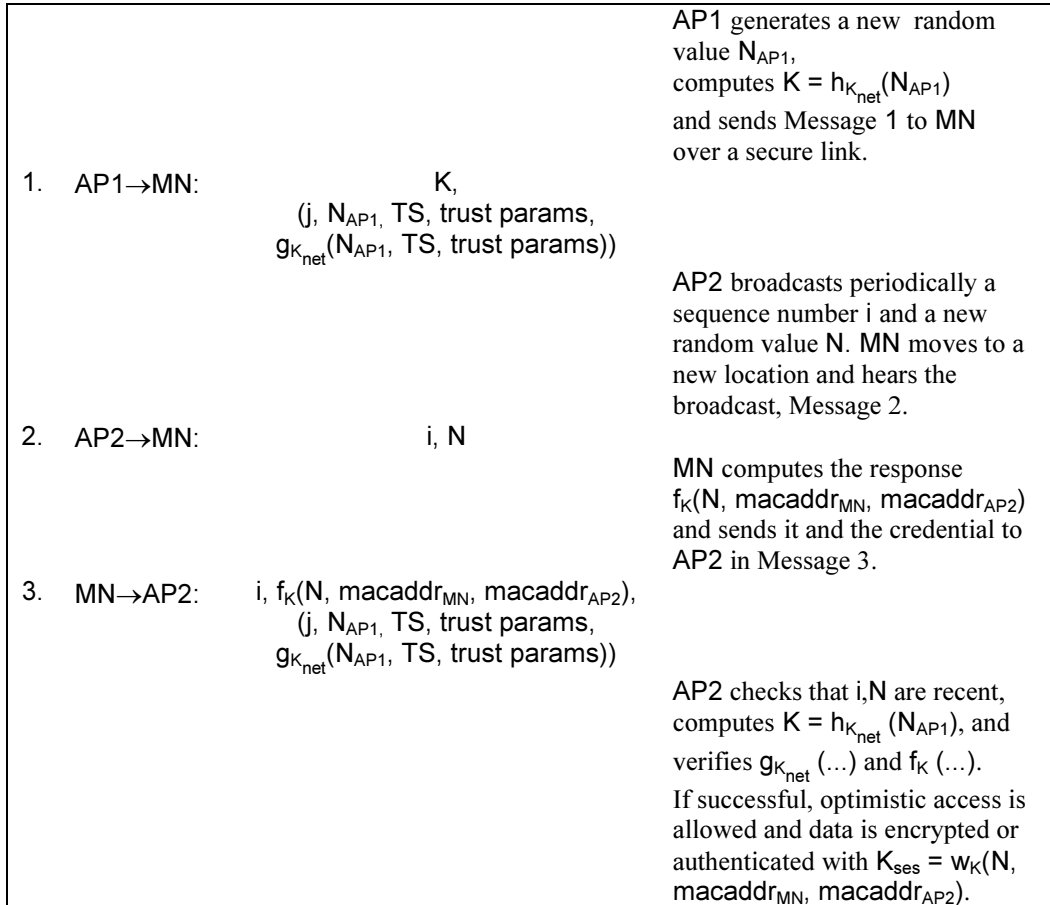


Figure 3 The complete protocol with one-way functions

$$K = h_{K_{net}}(N_{AP1}).$$

The keyed one-way function h used for the computation must be different from the g used for computing the response. The credential now needs to contain the nonce from which the key is computed instead of the key itself. Thus, the final form of the credential is

$$\text{Credential} = j, N_{AP1}, TS, \text{trust params},$$

$$g_{K_{net}}(N_{AP1}, TS, \text{trust params})$$

5. Complete reauthentication protocol

The complete reauthentication protocol is shown in Figure 3. After a successful execution of the protocol, the session key K_{ses} is a shared key between MN and AP2. It is then used for securing all traffic between them until a stronger session key becomes available. The access point should initiate a strong authentication or payment protocol but it should allow

the mobile to access the Internet without further delay.

Our credential is somewhat similar to a Kerberos ticket [9] in that it is an access credential protected by secret-key cryptography and it is presented to a service to obtain access. The differences are that our credential can be issued by any one of the co-operating servers, it can convey only authorization and does not need to contain an identity, it usually contain a record of the subject's past good behavior, and it is implemented purely with hash functions. Although there are considerable differences, we have also borrowed some ideas from the GSM security protocols [7], where inexpensive keyed one-way functions are used for both authentication and session key generation.

The keyed one-way functions can be implemented by computing the value of a secure hash function, such as SHA-1 [11], of the key and the message. The functions f , g , h and w can be implemented as follows:

Function	Purpose
$f_K(M) = \text{SHA-1}(1 K M K)$	response
$g_K(M) = \text{SHA-1}(2 K M K)$	credential signature
$h_K(M) = \text{SHA-1}(3 K M K)$	secret generation
$w_K(M) = \text{SHA-1}(4 K M K)$	session-key generation

Alternatively, the HMAC of Bellare et al. [3] can be used. It has stronger security guarantees than the construction suggested above but it requires the computation of two hash functions instead of one, thus doubling the computational cost of the protocol. If the function is implemented in hardware, something similar to the various derivatives of GSM COMP128 could be used. When choosing the hash function, it is useful to note that the hash function does not need to have strong collision resistance.

6. Trust parameters

The role of the trust params field in the credential is to encode information about the mobile node's past access to the network. The new access point will use this information as its basis for deciding how much it will trust the mobile node before it has strongly authenticated the node or received a payment. It is important to note that only positive information in support of the mobile node can be stated as nothing would prevent the mobile from hiding a negative credential from the new access point. For example, it would not make sense to issue a credential that says the mobile failed to make a payment.

A general guideline is that the parameters should record facts about MN's earlier network access rather than explicit authorization for future access. The credential could, for example, contain the time of the last strong authentication, the total payments MN has made for network access within a certain period, or a credit rating given to the mobile.

The credential should always be stamped with the time of issue or, preferably, the time when the facts were originally recorded. It would be less meaningful to state an expiration time or direct instructions about allowed access in the credential because that would prevent the new access point from implementing its own optimistic-access policy.

The new access point should make its judgment based on the facts in the certificate and any other information available at the time of verification. That way, the network operator may adjust the credential expiration times dynamically and search for the best balance between efficiency and security. The network

key K_{net} , which is shared by all the co-operating access points, may be replaced at any time, effectively revoking optimistic access for all mobiles until they have obtained a new credential. This allows the network operator to react to security breaches without waiting for credentials to expire, although we don't expect the feature to be used frequently. Moreover, some parts of the network may have stricter policies on the user authentication and advance payment than others, depending on the financial risk involved. Access points in these areas should be able to make their own decisions based on the facts recorded in the credential. For example, network areas with premium rates or high occurrence of fraud might honor only credentials that show recent payment at a local access point.

The trust parameters do not need to contain any global identifier for the mobile node but, technically, one can be included. In that case, the identifier should be encrypted with K_{net} and using N_{AP1} as the initialization vector so that the identifier is not revealed to outside listeners. This, however, increases the complexity of the protocol. An alternative is that the access point issuing the credentials logs the nonce N_{AP1} with the related identity or payment information. The log data can be used off-line for analyzing the events if, for example, it is found that the mobile is misusing the optimistic access or distributes the credential to others. The appropriate reaction to such misuse would probably be to revoke the strong authentication credentials but that is beyond the scope of our protocol.

7. Efficiency

The critical part of the protocol from a performance point of view is the reauthentication between MN and AP2 (Messages 2 and 3). The duration of this exchange determines the delay before the mobile can access the network through the new access point. The computational cost of our protocol is very reasonable, especially for the mobile node. The mobile needs to compute the response and the session key, which takes two evaluations of a keyed one-way function in total. The new access point needs to perform four such evaluations: the first to obtain the key K , the second to verify the credential, the third to verify the response, and the fourth to compute the session key. This is still only a modest amount of work given that access points can be quite powerful computers and are powered by mains electricity. The speed of the setup phase between AP1 and MN is not critical as long as it completes before the mobile moves away from the first access point. The additional computation required by our protocol is

	Computation Setup (AP1-MN)	Reauthentication (MN-AP2)	Memory
AP1	strong authentication, g, h		K_{net} , no per-mobile state
MN	strong authentication	f, w	K , Credential
AP2		f, g, h, w	K_{net}, N , no per-mobile state

Figure 4 Computation and memory cost

Message	Bits over air
1	$64 + (64 + 64 + 64 + \text{trust params} + 64)$
2	$16 + 64$
3	$16 + 64 + (64 + 64 + 64 + \text{trust params} + 64)$

Figure 5 Communication cost

negligible compared to any strong public-key authentication.

The access points remain stateless until the reauthentication has completed. They do need to remember the shared key K_{net} and at least the two latest values of the broadcast challenges N while they are valid. Since the same values of K_{net} and N are used in the authentication of all mobile nodes, the additional cost for each new mobile is zero. The mobile node needs to store the key K and the credential. The computation and memory costs of the protocol are summarized in Figure 4. The low computational cost of the one-way functions and the statelessness of the access points are essential because they protect the access points against denial-of-service attacks that attempt to consume processor and memory resources.

The protocol messages are relatively short. If keys and one-way functions with 64-bit values are used, the counter i is 16 bits, and the key identifier j and timestamp TS are 64 bits each, the total number of bits transferred over the radio interface between the mobile and access points is approximately 736 plus two times the size of **trust params**. The lengths of the individual messages are listed in Figure 5. Overall, the messages are still short compared to ones needed in public-key protocols. Note that the keys and hash values do not need to be very long because the window of attack is typically only a few seconds.

An additional task for the access points is to generate the nonces N_{AP1} and N . These should be unpredictable random or pseudorandom numbers. The mobile does not generate any nonces.

From this we see that the protocol is suitable for mobile devices with very limited computational and

communication capacity. It is possible to further optimize the implementation by using hardware-based one-way functions and variable-length fields for the values, where the lengths should be decided by the access points depending on the current level of fraud in the network.

8. Security details

The shared-key protocol is efficient but it also has obvious limitations, which we will consider in this section. The two most important ones are that the access points must share a secret key and that the mobiles are trusted not to share or publish their credentials. We will also discuss replay attacks that might pose serious threats in other applications but do not prevent our protocol from achieving its goals.

The access points of the network share a secret key and must trust each other not to misuse it. If a single access point leaks the key or issues credentials with less care than the others, the entire network will suffer. Moreover, it is not possible to securely identify which access point issued a credential. The credentials should probably contain an access-point identifier of the issuer but that will only help in identifying accidentally misconfigured access points, not malicious ones. Partially for these reasons, we are not suggesting to replace the 3strong authentication entirely. Instead, our protocol should be used for reauthentication within a limited period and geographical region from the last strong authentication or payment, and even then only to allow optimistic access to the network for the time that it takes to perform the strong authentication. Another mitigating factor is that the credentials

contain the key index j , which allows an access point to belong to several co-operating groups. There can be multiple separately managed and overlapping coalitions of access points or networks so that, if the secret key for one is compromised, the others can continue to operate. The shared network key K_{net} should be replaced periodically and whenever key compromise is suspected. The effect of a key change for the mobiles is that all credentials are rendered invalid and all mobiles must be reauthenticated. The key index makes it possible to allow multiple keys in parallel and to update the network key without the need for simultaneous reconfiguration of all access points.

Besides the shared network key, another weakness of the system is that nothing prevents a mobile node from sharing its credential and secret key K with other mobiles. It is important that such malicious mobiles can be identified. Therefore, the access point issuing a credential should log the mobile identity (i.e., any identifier used in the strong authentication or authorization) for the expected or maximal validity period of the credential. Another way to discourage sharing of credentials would be to include a hash of the mobile identity and a random salt in the credential. The new access point would always ask for the identity and check that it matches the hash. The mobiles that repeatedly publish their credentials and secret keys could then be caught. However, this would mean that the mobile identity is always revealed to all access points.

The inclusion of the MAC addresses in the arguments of the one-way functions f and w can be seen as an application of *hashed full information* [2]. The idea is that all information that is specific to the protocol run and shared by the receiver and sender of a message should be included in a hash value in each message. The cost is often negligible but the chances of the protocol having hidden flaws is significantly smaller. In our protocol, the inclusion of the MAC becomes important if the communication during the optimistic access period cannot be encrypted, as we will explain next.

The protocol uses two mechanisms for preventing an attacker from performing replay attacks and from hijacking a reauthentication completed by another mobile. Firstly, if the data sent during the optimistic access period is encrypted or authenticated with K_{ses} , the attacker can only replay packets sent by the honest mobile. The attacker does not win anything by doing that because the goal of the attacker in the scenario of this paper is to gain access to the network, not to mount replay attacks against other mobiles. (End-to-end security mechanisms and even TCP duplicate detection can be used for protecting the data against replays.) Secondly, whether the data is encrypted with

K_{ses} or not, Messages 2-3 of our protocol bind the network access to specific mobile and access point MAC addresses. Thus, the attacker can only access the network by observing a reauthentication and then spoofing its MAC address to be the same access point to which the honest mobile is connected. Again, the attacker gains very little because the honest mobile will be trying to access the same access point using the same MAC address during the same short period of optimistic access. Thus, even without any encryption, the attacker can only mount a DoS attack against the honest mobile; it cannot obtain reliable network access for itself although a few packets might get through. Hence, the attacks that in other situations might be critical, are not serious threats in against our protocol.

The challenge-response mechanism is not strictly necessary to prevent a dishonest mobile from accessing the network during the optimistic access period. That is, if the key K_{ses} is used for encrypting or authenticating the traffic on the wireless link, that is sufficient to prevent the mobile from benefiting from the network access unless it knows the session key. The response does, however, protect the new access point from denial-of-service attacks where the attacker creates many optimistic associations and never uses them. It also provides some security on links where the session key is not used for link encryption or authentication.

The keyed one-way functions g and h may be optimized for efficiency but f and w should be selected with care and long-term reliability in mind. The reason is that f and w are implemented at the mobile node while the other two functions are internal to the access network. The network may change its internal implementations relatively flexibly but it is usually impossible to update all mobile nodes. It is nevertheless possible to parameterize the lengths of the values of all the one-way functions, as long as the network operators decide the length that is currently used.

While the protocol supports anonymity for the mobile by not requiring any mobile or user identity to be included in the messages, it does not specify what information the access points may insert into the credentials. Since the credential is usually transmitted in plaintext, its contents can be observed by anyone listening to the communication channel. Furthermore, privacy concerns similar to ones related to HTTP cookies may arise with the credentials. Plaintext credentials can, in fact, be an advantage to privacy-conscious mobile users because they are able to inspect the contents either every time before forwarding the credentials to the new access point or only occasionally to detect deviations from a privacy policy. There will, however, always be room to hide

some compromising data, for example, in the nonce N_{AP1} .

9. Conclusion

In this paper, we describe a mechanism for fast reauthentication of a wireless mobile node when it returns to the same access point or to another, cooperating, one. The protocol is a deliberate compromise between security and efficiency and it is intended to only enable optimistic network access before a strong authentication of the mobile takes place. The protocol has the potential to reduce significantly authentication delays experienced by mobile nodes at each new access point that they connect to and, thus, to improve the quality of service experienced by mobile users. This is important because security protocols are currently a major source of latency in mobile and wireless communication.

References

- [1] Tuomas Aura, Pekka Nikander, and Jussipekka Leiwo. DOS-resistant authentication with client puzzles. In Proc. Security Protocols Workshop 2000, Cambridge, UK, volume 2133 of Lecture Notes in Computer Science, pages 170-177. Springer.
- [2] Tuomas Aura. Strategies against replay attacks. In Proc. 10th IEEE Computer Security Foundations Workshop, pages 59-68, Rockport, MA USA, June 1997. IEEE Computer Society Press.
- [3] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Keying hash functions for message authentication. In Advance in Cryptology - Crypto 96 Proceedings, volume 1109 of LNCS, pages 1-15, Santa Barbara, CA USA, May 1996. Springer.
- [4] CCITT. Recommendation X.509, The Directory - Authentication Framework, volume VIII of CCITT Blue Book, pages 48-81. 1988.
- [5] David Chaum, Amos Fiat, and Moni Naor. Untraceable electronic cash. In Advances in Cryptology - Proc. CRYPTO '88, volume 403 of LNCS, pages 319-327, Santa Barbara, CA USA, August 1988. Springer.
- [6] Jon Edney and William A. Arbaugh. Real 802.11 Security: Wi-Fi Protected Access and 802.11i. Addison-Wesley, 2003.
- [7] ETSI TC-SMG. GSM 03.20 (ETS 300 534): European digital cellular telecommunications system (Phase2); security related network functions, September 1994.
- [8] ANSI/IEEE Std 802.11. Information technology -- telecommunications and information exchange between systems -- local and metropolitan area networks -- specific requirements -- part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, 1999.
- [9] John Kohl and B. Clifford Neuman. The Kerberos network authentication service (V5). RFC 1510, IETF Network Working Group, September 1993.
- [10] Catherine Meadows. A formal framework and evaluation method for network denial of service. In Proc. 12th IEEE Computer Security Foundations Workshop, pages 4-13, Mordano, Italy, June 1999. IEEE Computer Society.
- [11] National Institute of Standards and Technology (NIST). Secure hash standard. Federal Information Processing Standards Publication FIPS PUB 180-1, Gaithersburg, MD USA, April 1995.
- [12] Carl Rigney, Steve Willens, Allan C. Rubens, and William Allen Simpson. Remote authentication dial in user service (RADIUS). RFC 2865, IETF, June 2000.
- [13] John R. Vollbrecht, Pat R. Calhoun, Stephen Farrell, Leon Gommans, George M. Gross, Betty de Bruijn, Cees T.A.M. de Laat, Matt Holdrege, and David W. Spence. AAA authorization framework. RFC 2904, Network Working Group, August 2000.